

New General-purpose Tools for Getting CU Data Into CU*BASE:

DRAFT

- ▶ CU-directed data floods
- ▶ General-purpose upload files

October 20, 2016

Spec Writer: Dawn Moore

Systems Affected:

- CU*BASE

Overview

This project makes it easier than ever for CUs to pull data from an external source into CU*BASE, both for analysis as well as for manipulating common CU*BASE data elements (commonly referred to as “floods”).

There are two main components to this spec:

1. Creating a new tool that allows a CU to direct their own data routines for common data elements
2. Facilitating uploads of common data elements, such as account numbers or email addresses, into generic CU*BASE files

The second component is a requirement for accomplishing the first part (that is, allowing CUs to pull in the list of accounts they want to use in their flood), because of the way the IBM i requires data to be uploaded into a file that already exists and is formatted to receive the data.

But it will also facilitate CUs pulling in lists for other purposes, such as Common Bonds or Member Connect functionality, or for use with CU*BASE analysis dashboards. For example, the new Analyze Member Behavior Patterns (a.k.a. “Losing the Love”) dashboard allows a CU to enter a file name in place of using the standard filters to select which accounts will be analyzed.

(Doc writer see Page 7 for suggested content for the release summary.)

Excerpt from the 2016 Data Investment Symposium

Contents

Project Overview.....3
 Part 1: CU-Directed Data Floods3
 Part 2: General-purpose Files for Uploading Data to CU*BASE4
 Data Validation and Controlling Upload Security5
Suggested Content for the Release Summary7
 Run Your Own Data Floods for FREE7
 Upload Any Account List for Data Analysis or Floods.....7
 Miscellaneous Doc Notes.....8
Summary of Program Changes 10
Flood Types 11
 Flood Types to be Included in Phase 1 12
 Ideas for Future Phases (Later) 15
New Master Flood Types Configuration 17
New Database Flood Engine 18
 Screen 1 - Choose Flood Type 18
 Screen 2 – Choose Flood Settings 19
 Screen 3 – Validate & Process 21
 Reports 22
Examples of Existing Flood Programs..... 23
 MNMRKT #7 Auto-Enroll Marketing Club Mbrs 23
 MNMRKT # 10 Tag Mbr Accts with Letter Codes 24
 MNOP21 # 1 Flood e-Stmt Mbrs for e-Notices 25
 MNOP21 # 2 Flood Mail Code to Opt Out Flag 26
 MNOP21 # 3 Credit CARD Act--Flood Due Dates 27
 MNOP21 # 4 Gender Generator 28
 MNOP21 # 5 Open End Contract Flood 28
 MNOP21 # 9 Credit Score Soft Pull - Upload 29
 Reassign Tracker Follow-ups..... 29
 Wrong Email Address Flag & Marketing Opt In/Out Flags..... 29
 Others currently in Toolbox?? 30
Menu Changes 30

Project Overview

Part 1: CU-Directed Data Floods

Frequently we get requests from credit unions who want us to flood certain values into a batch of member or account records in CU*BASE. Examples include updating loan review dates on all loans of a certain category, or flooding negative balance limits on a batch of checking accounts to launch a new ANR program.

Traditionally these floods are handled as custom projects for which the CU is charged a development fee, and we work directly with the CU to process the flood according to their criteria. Beyond putting the new values into the database for a batch of records, we also assist with setting up the criteria that determines which member and accounts records will be changed. And of course our programs also include validations to ensure that no invalid or garbage data gets written into live CU*BASE member files.

Letting CUs Perform Floods Themselves

This project creates an engine that will allow credit unions to take care of the most common types of floods by themselves, without having to involve a programmer.

But unlike a custom program that would handle both the selection criteria and the values to be flooded, the idea here is that the credit union employee first would use other existing tools – such as Query or Member Connect exports – to develop the list of accounts that should be flooded. From there the new engine would take care of performing the flood to that list of accounts, including any validations we deem necessary to ensure only good data is being written to the database.

Two Flood Types: Flood Generic Value vs. Use Value from File

❶ The first and simplest type of flood would be for data elements that have consistent, pre-defined values, such as loan purpose and security codes, SEG/Sponsor codes, simple on/off (Y/N) flags, and the like. Once the list of eligible accounts is ready, the credit union employee would access the new tool, choose the table column (field) to be flooded, and specify the generic value to flood into all the accounts in their list.

Example: An employee uses an Export to Member Connect feature on the Credit Score Analysis dashboard to gather a list of account numbers for members who have a credit score below 600. He then takes the resulting list of account numbers and runs it through the data flood routine to change the Electronic Deposit Hold Group to 03 on all of those members.

❷ As a variation, instead of flooding a generic value to all of the accounts, there will also be an option to pull the new values directly from a value in the credit union's file, rather than using a generic value for all accounts. In addition to gathering the data from the file, the system would perform validation to ensure that invalid data was not populated into the CU*BASE database.

*Example: An employee downloads a list of accounts and uses Excel to organize and refine the list, then adds an additional column with values they want to use for a member's ANR limit. The list is uploaded back to CU*BASE and used by the flood routine to populate the Negative balance limit on each of the accounts in the list, using the limit value the CU specified for each individual account.*

Posting and Exception Reports

In both cases, the process will produce a **posting report** to show the old and new values after the update, and an **exception report** to show any account numbers from the credit union's list that could not be updated for any reason. There will also be an optional "audit" routine that will allow the CU to run a

test of the flood process without actually updating any accounts, so they can review the reports to verify the results before actually performing the maintenance.

Technique: Building an Engine for Future Floods

Beyond the basic structure outlined in this spec, I'd like to develop a behind-the-scenes configuration that makes it easier to add additional data elements to the flood tool down the road. Instead of doing specific hard-coding for each flood type we can think of now, we'll create a new database with a maintenance tool that allows a programmer to enter field names and validation routines into a table, and then design the end-user screens to be able to adapt on the fly to new items as they are added.

Much like the 5300 Call Report structure, this would allow us to quickly adapt to new CU needs without the burden of having to coordinate every change around a CU*BASE software release. Although it might increase the development time somewhat, we'll get a better long-term product in the end, one that will better meet the needs of CUs in the future.

Part 2: General-purpose Files for Uploading Data to CU*BASE

In order to handle the feature in the new tool described above that allows the CU to give us the list of account numbers to be flooded, we need a way for the data to be uploaded into CU*BASE.

Because of how a System i works, data must be pulled into an existing file on CU*BASE, large enough to contain the data being uploaded. The upload process itself **requires an *.FDF (Format Description File) on the PC**, which describes the format of the incoming data. As with the other standard imports we offer via CU*BASE tools (general ledger/budget records; Direct/Mail Post member transactions; match lists from FinCEN, OFAC, and FIDM; Membership Applications, etc.), we will deploy the associated *.FDF files with the CU*BASE GOLD package to credit union GUAPPLEs so that they are available on any workstation that uses CU*BASE.

Therefore we will be creating several general-purpose files into which a credit union could upload data from a PC into the System i for flooding, or for use with CU*BASE analysis tools. For example, some CUs want to be able to download member names and information to a PC program like Excel and do various manipulations with it to compile a list of members with some particular characteristic. Then they want to upload the finished list of account numbers back to CU*BASE for various purposes:

- Join with other files – Credit Scores, Credit score trending, ACH deposits, Tiered Services, Loan/Deposit balance, Where Your Members Borrow data, etc.
- Leverage the Common Bonds features
- Create Member Connect touch points – Telemarketing trackers, Online Banking messages
- Add Next Suggested Product
- Marketing Club auto-enrollment
- Statement selective inserts
- ...and of course to use with the new CU-directed flood tool being created in this project

Once the data is pulled into the file, Query can use it as is, just by joining that file to any other file on the system, like MASTER or MEMBERxx. Assuming the file has a 9-digit numeric account base in the first column, Common Bonds and Member Connect features can also use the file as is.

NOTE: If we also wanted to allow CUs to take data from that file and actually use it within a more complex CU*BASE feature, like we did with Direct/Mail Post or the G/L JE Posting system, we'd have to create another program to **convert the data in the new file into a**

compatible format for that particular program. For the new flood tool already mentioned, we'll be building that into the tool, as described later in this spec.

For other places where that functionality is needed, we would add an Import button to the CU*BASE screen where they need to be able to use the data, and incorporate some simple data validation scans (numeric data formats, date formats, etc.) to prevent any potential conflicts with that program. For now, other than the new flood tool, these will be the exception, not the rule, so we'll deal with these on a case by case basis as needed in the future.

Proposed File Formats

Although we will add more down the road once we start getting some feedback from our power users, for this first phase we'll include four formats that will likely cover the majority of situations CUs are encountering today:

<i>Type</i>	<i>Suggested Table Name (see NOTE below)</i>	<i>Contents</i>	<i>Purpose</i>	<i>Validation</i>
Account or Tax ID #	IMPMBR#	One numeric field, 9 digits #####	For bringing in a list of memberships or non-member SSNs	Must be numbers
Account base and suffix	IMPACCT#	Two numeric fields: 9 digits and 3 digits ##### ###	For bringing in a list of member sub-account numbers, base and suffix	Must be numbers for both columns
Account base and free-form text	IMPTEXT	One 9-digit numeric field, followed by one 300-byte alphanumeric field ##### ABC1234xxxxx 300	Examples: for bringing in a file of email addresses tied to a member account base or SSN/TIN (for members or non-members), or for populating comments or other free-form text fields, up to 300 characters	Must be numbers for the first column

Misc. notes:

- All tables will use DDS. *(Per Deb Stickler: Our current standards are DDL tables. The data had better be PERFECT or the DDL table will throw it out. If it's an API, they can do the editing on their side. If it's CA upload, then we had better make it a DDS file. DDS file does not confirm the data until the read.)*
- Table names to be assigned per our normal procedures; these are suggestions only but would like to keep the IMP* format if possible, even though it doesn't match our existing ones which are GLIMPORT, DMIMPORT, and MBRIMPORT (plus FCPERSON and FCBUSINESS for FinCEN).
- We'll need to make sure the files that get put into QUERYxx are set up so they don't accidentally get purged if the CU doesn't use them regularly. I understand there's a parm you can put on the tables?

Data Validation and Controlling Upload Security

Our goal is to make it easy to pull data into CU*BASE, while still maintaining tight controls over data that could potentially affect the live member data we are obligated to protect. We don't want an upload to inadvertently cause an error in EOD or corrupt any of the CU's live data!

As now, uploads will be allowed into the QUERYxx library ONLY. CUs who need to pull data into a table stored in FILExx will need to perform a second step once the data has been uploaded. Examples:

- To use an external list to populate one of the **selective statement insert** files FILExx/STMT##, after uploading the data to a location in QUERYxx, the user would run a Query to output those account numbers into the STMT## table in the FILExx library.
- Our existing import tools already mentioned require specific programs to be run that automatically **validate the data and convert it** for use by CU*BASE formats. For example, the budget import process ensures that G/L accounts in the incoming file match what's in the credit union's chart of accounts and that alphanumeric data isn't accidentally fed into fields with numeric or date formats, etc.

As far as **upload permissions** go, CU staff must still [request authorization](#) to upload into their QUERYxx library. In the past the authorization was strictly by table name – so a person had to request access to the QUERYxx/GLIMPORT file, and separate access to the QUERYxx/MBRIMPORT file, etc. With this project, for staff members who need to perform many different uploads to many different tables, we will expand the authorization form to allow a CU to request upload authority to all tables in QUERYxx, if desired. CUs will still be able to limit a person's access to a single file where appropriate, as well.

Suggested Content for the Release Summary

This should be directed at Credit Union Data Analysts – suggest we start having a special section of the release summary for them, given our new emphasis on data in this year’s business plan.

Run Your Own Data Floods for FREE

If you’ve ever paid for custom programming to flood certain values into a batch of your member accounts, you’ll love this new Do-It-Yourself tool! The new **XXXXXX** tool performs many of our most commonly-requested flood routines. It has built-in audit and exception reports, with validation to ensure only clean data is written into your member files. You can even use a file you’ve uploaded* to determine the values to be flooded, if you like, or choose a value from a configuration.

Of course there’s no way for a generic tool to handle every possible flood scenario, and there will always be requests that will end up requiring custom programming. But with **xxx** different flood routines to choose from, this tool will go a long way toward handling the more common requests, saving you money and adding flexibility to how you manage your member data.

*NOTE: This tool requires you assemble a list of account numbers, which can be done via the CU*BASE Report Builder, the List Generator, or any of nearly 100 analysis dashboards that have export functionality. You can also upload a list you’ve assembled in Excel or another PC program and upload it into CU*BASE, using the new generic upload files being introduced with this release. See below for more details.

Upload Any Account List for Data Analysis or Floods

Now it’s easier than ever to pull in a list of accounts created in Excel or another PC program and upload it into CU*BASE for use with CU*BASE Report Builder or List Generator for data analysis, or with the new **flood tool** already described.

The tricky part of uploads is that there already must be a place to put the data on the System i and a *.FDF file on your PC to handle the data formatting. With this release we’ve created several new generic database tables into which data can be uploaded into your credit union’s QUERYxx library, and will include the *.FDF files in the CU*BASE GOLD software that’s on your workstation. All you’ll need is to request authority to the files, and you’re ready to pull data into:

<i>Table Name in QUERYxx</i>	<i>Contents</i>	<i>Purpose</i>
IMPMBR#	One numeric field, 9 digits for an account or tax ID number #####	For bringing in a list of memberships or non-member SSNs
IMPACCT#	Two numeric fields: 9 digits for an Account Base and 3 digits for an Account Suffix ##### ###	For bringing in a list of member sub-account numbers, base and suffix
IMPTEXT	One 9-digit numeric field for an Account Base, followed by one 300-byte alphanumeric field for any text ##### ABC1234xxxxx...300	Examples: for bringing in a file of email addresses tied to a member account base or SSN/TIN (for members or non-members), or for populating comments or other free-form text fields, up to 300 characters

These new tables join existing ones already in place for imports via Direct/Mail Post, Budgeting, G/L Posting, Membership Applications, and others. Refer to ??? for more details on all of our file upload capabilities. (We should have a summary help topic about this if we don't already – see the 2016 Data Investment Symposium presentation slides 14+ and 2015 Data Investment Symposium presentation slides 52+.)

Also add note about how they have to fill out the iShield authorization request form to get permission to do these uploads. More below.

Miscellaneous Doc Notes

- Make it clear to CUs that in order for their staff to use the new tables they will need to submit a new authorization form, requesting either the specific tables they want to use, or the new “all” authority. **(Per Todd Wolcott the “all” authority would require us to delete the user’s existing authority in iShield and recreate it with the broader authorization.)**
- Also reiterate that it is the CU’s responsibility to monitor the daily and monthly reports we produce (daily report LSECAUDIT shows who downloaded/uploaded something and monthly report LSECAUDITM shows who has download/upload authority) in CU*Spy.
- In addition to floods, uploads can be used for analysis. CUs will either use Query to join this new table with other CU*BASE files for custom reports/inquiries, or simply use the file name in connection with CU*BASE Member Connect, Common Bonds, or other similar tools.
- Imports will overwrite whatever is already in that file. Be sure to coordinate the timing with anyone else at your credit union who is authorized to upload data into files in your QUERYxx library. CUs with busy data analyst teams may wish to use the import table as only a temporary holding place, immediately employing a Query to output the data into a separate table to be used for their final analysis needs. (Maybe show the steps for doing this, recommend saving that as a simple “translation Query” definition that they can use over and over.)
- Explain how alternatively they could use the IBM Client Access Upload Wizard to also create their own, custom *.FDF file to import data and create the table on the System i at the same time. Be sure to show the Details button on the “Contents of PC File” window that lets them change the assigned fields of F1, F2, F3, etc., to something meaningful (this same feature also lets you expand the length for fields that might need to be different from whatever the longest item is in the incoming file).
- Publish a tip about how if you link the file to MASTER (or MEMBERx), you can use the join type 1 to see all the hits, but then simply change that query to join type 3 to see any records you uploaded that did NOT get a hit on MASTER. This would be useful in cases where they are pulling in a file but won’t be using one of the tools that gives them a list of hits/exceptions.
- Remember that you can already upload a prospect list (potential members) into the MBRIMPORT file, using the procedures outlined in the [Importing a List of Prospective Member Names into the CU*BASE Pending Membership Applications Database](#) booklet.
- Remember that the CU’s file must be in QUERYxx (uploads direct to FILExx are not allowed). (We monitor disk usage and manage the tables in QUERYxx and any CU-created tables are automatically purged after 3 months of non-use. The standard ones described in this spec, though, are set up so they don’t get purged even if they aren’t used, so they’re always ready to go for when the CU decides to start using them.)
- Since the new engine has baked-in validation routines and controls which fields can be flooded, any additional floods not included can be requested either as a custom (\$\$) project, or as an enhancement for the flood engine, handled like any other project request via the SDLC process.
- Make sure to give a comprehensive list of all the ways they can gather a list besides Query. Don’t have to list specific dashboards, necessarily, although could certainly point to the

Leadership Dashboards brochure and explain that the “Export to Member Connect” (or whatever the button/option reads) is how they would gather the list of accounts. Other things to mention:

- MNMRKT #11 Random Mbr Analysis/File Build
- MNMRKT #12 Aggregate Analysis/File Build
- Any file received from a third party as long as it is has ACCTBS and, if appropriate, ACTTYP as columns #1 and #2, respectively
- Tip for if they can't identify the list of members any other way – Letter Codes (MNMRKT #10) provides an easy way to tag individual member accounts that do not have anything else in common, and then you could create a Query based on the value in that letter code field and use that file to run the flood. (One example for why to use letter codes is a contest where people put their names into a bowl for a drawing, and you want to be able to get a list of all those members and do something with them via Query – you could also then use the list for a flood.)

Summary of Program Changes

The following program changes are anticipated with this project:

- Create new files in all QUERYxx libraries with corresponding *.FDF files to be included in the GOLD update package, similar to how we do others (GLIMPORT, MBRIMPORT, DMIMPORT, etc.). See the chart on Page 5 for specs.

NOTE: Remember the purge process for QUERYxx – don't want to accidentally purge these just because the CU hasn't used them yet. I understand there's a parm that can be put on the table when it's created to prevent that?

- Create a new Master Flood Type config/maintenance engine and file structure to support the new flood tool, so that we can add additional fields/validation routines in the future without having to do hard-coded programming work. Maybe something similar to how we designed the 5300 account code maintenance tools? **This part needs to be designed by the programmer.** Please reach out to the spec writer with your ideas and we can talk this through.
- Create the new Database Flood maintenance tool – see Page 18
- Create a posting/auditing report and an exception report – see Page 22

New CU*BASE Tables

The following decisions must be considered if any new tables are needed with this project:

Table name (use standard naming convention)	
Description - how will the table be used?	
Special table specifications	
EOM table required?	
Purge criteria to ensure the table is kept clean: Does the table need a daily and/or monthly purge? Can records be purged when the account/membership closes or other activity is done?	
Does this table obsolete any other tables?	
Sign-off by Bob C	

Evaluation for Stand-in Processing

Decisions that must be made if this new/enhanced feature will affect **It's Me 247** online banking, CU*Talk Audio Response, or EFT (ATM/Debit/Credit card) processing in stand-in mode.

n/a

Flood Types

A = CU provides a list of accounts, screen asks for a generic value to plug into the flood field (no config validation)

Use this for flooding codes where there is a predictable value such as Y/N, 0-9, 00-99, etc. but no CU-defined configuration against which the value must be validated

Should we also include a variable on the flood screen so they can have a list of account base numbers only and then specify which account types should be flooded? I.e., choose loan category codes/DivApl/CD Types, or maybe just "all loans" and "all savings/checking accounts" or something?? Or see type "D" below?

Example was to be able to get a list of mbrs with credit score below 500 (from the credit score dashboard) and use that list of memberships to change the Collector ID on all of their MEMBER5/6 loan accounts. Otherwise they'd have to first get the list of members, then somehow generate a file of all loan accounts under those memberships before they do the flood.
Maybe wait for phase 2?

B = CU provides a list of accounts, screen offers a lookup and asks for a value to plug into the flood field and validates the entry against a CU or system config

Use this for flooding codes where there is a configuration for valid codes, like Electronic Deposit Hold groups, SEG/Sponsor codes, Marketing Clubs, G/L account numbers, etc.

Include an "all accounts" option? (see note above)

C = CU provides a list of accounts and new values for the flood field, screen validates and gives stats on what's going to be done but uses the data in the file to actually populate the flood field

Particularly important to have an exception process on this one that shows any of the data that couldn't be flooded (i.e., dates in the wrong format, dollar amounts too large, that sort of thing)

D = CU provides a list of accounts and new values, screen validates but also provides additional options/conditions to refine the flood - **SAVE FOR PHASE 2**

Don't have any specific examples of this type yet, but maybe this is where we could put simple suffix-level selection criteria so that they only have to provide a list of members and then use the tool to refine which sub-accounts to flood by selecting DivApl(s), LnCatg(s), or CDTyp(s).

All flood types should validate against MASTER and show # of records that will be updated. The posting/audit report should show old and new values as the updates are made. Also need some sort of exception reporting mechanism if records are encountered in the CU file that can't be matched to a valid record in the CU*BASE files.

So far I haven't included any floods of non-member or closed history. Need to understand the list of completed floods a little better to see if there are any common ones we should also include. Considered allowing all three files to be used for any floods where the values apply (i.e., where the same field appears in all of those files), and just make them specify, but it got rather complicated and hard to think through. **SAVE IDEA FOR PHASE 2**

Flood Types to be Included in Phase 1

Flood Type	Data Element to be Flooded	Required in the CU's File	Flood Field Options and (Format Validation)	Validation to Perform	Other Options on the Flood Screen?
A	Additional Signers Delinquency Level	Loan ACCTBS Loan ACTTYP Add'l signer SSN	<i>New Delq Level # (0-4)</i>	Screen edits only	
A	AFT Loan Payment Controls	Loan ACCTBS Loan ACTTYP	<i>Delq Ctrl (Y/N)</i> <i>Pmt Ctrl (Y/N)</i>	Screen edits only	
A	ANR Limits and Lock Flag (standard)	ACCTBS ACTTYP	<i>Neg Bal Limit (\$9.2)</i> <i>Neg Bal Lock (Y/N)</i>	Screen edits only Allow for "leave as is" option for both	also see separate entry for CU-defined limits
C	ANR Limits and Lock Flag (from list)	ACCTBS ACTTYP Limit (9.2) Lock (Y/N)	<i>Neg Bal Limit (\$9.2)</i> <i>Neg Bal Lock (Y/N)</i>	Allow for "leave as is" option for both so they could have a file with just one or the other value Check for valid data in the file; put on exception report any \$ amounts that are too large, or if anything other than Y/N in the lock field	also see separate entry for standard limits – will combine these on the same screen, I think – see Page 19
A	CD Penalty Code	Cert ACCTBS Cert ACTTYP	<i>New CD penalty code (L, G, or blank)</i>	Screen edits only	
A	Due Diligence Flag	ACCTBS	<i>New DD flag (0-9)</i>	Screen edits only (can't really validate against the config MNCNFC #19 since it's just a list of descriptions for the hard-coded numeric codes 0-9, and everyone has at least a default desc)	
A	ECOA Code	ACCTBS ACTTYP	<i>New ECOA code (0-9)</i>	Screen edits only	
B	Electronic Hold Group	ACCTBS	<i>New Hold group code (01-99)</i>	Validate against Hold Group config (MNCNFD #20)	
A	Freeze Code	ACCTBS ACTTYP	<i>New freeze code (0-3)</i>	Screen edits only	
B	Loan Assigned Employee ID(s)	ACCTBS ACTTYP	<i>New approval ID (##)</i> <i>New Interviewer ID (##)</i> <i>New Collections ID (##)</i> <i>New Opened by ID (##)</i>	Validate for valid Employee ID in SECMAS	(can choose one or all of the flood items; make sure to include a "leave as is" option so it's clear the IDs to leave alone)
B	Loan Delinquency Fine Codes	ACCTBS ACTTYP	<i>New delq fine code (00-99)</i>	Validate against Fine code config (MNCNFF #2)	

Flood Type	Data Element to be Flooded	Required in the CU's File	Flood Field Options and (Format Validation)	Validation to Perform	Other Options on the Flood Screen?
A	Loan Maturity and Review Dates	ACCTBS ACTTYP	<i>New maturity date (CCYYMMDD)</i> <i>New review date (CCYYMMDD)</i>	Must be valid date	<i>For both dates, can specify CCYYMMDD, or current date + xx months, or no change</i>
B	Loan Payment Matrix	ACCTBS ACTTYP	<i>New fine amt setting (0-4)</i> <i>New interest due setting (0-4)</i> <i>New principal setting (0-4)</i> <i>New principle amt (11.2)</i> <i>New escrow setting (0-4)</i> <i>New misc G/L proc (0-4)</i> <i>New GL acct (5.2)</i> <i>New GL amt (9.2) or % (5.2)</i> <i>New overpmt indicator? (XXX)</i>	Screen edits to make sure sequence #s aren't repeated and no sequence # is missed Mirror screen ULNPMU-01	
B	Loan Purpose/ Security Codes	ACCTBS ACTTYP	<i>New purpose code (00-99)</i> <i>New security code (00-99)</i>	Validate against purpose and/or security code config (MNCNFB #3 or 4)	(can choose one or all of the flood items, make sure to include a "leave as is" option so it's clear the IDs to leave alone)
B	Loan Variable Rate Code	ACCTBS ACTTYP	<i>New variable rate code (0-9)</i>	Validate against variable rate code config; account type must be a loan	
A	Marketing Opt In/Out Flags	ACCTBS	<i>New 3rd-party marketing opt out setting (Y/N)</i> <i>New CU marketing opt-out setting (Y/N)</i>	Screen edits only	
B	Member Branch	ACCTBS	<i>New member branch (##)</i>	Validate against location config in chart of accounts (MNGELE #2)	
B	New Account Reason Codes	ACCTBS ACCTYP	<i>New reason code (##)</i>	Validate against config (MNCNFC #1)	
B	New Member Reason Codes	ACCTBS	<i>New reason code (##)</i>	Validate against config (MNCNFC #1)	
A	Photo ID on File Flag	ACCTBS	<i>New Photo ID flag (Y/N)</i>	Screen edits only	
A	Reg E Opt Out	ACCTBS	<i>New Reg E flag (Y/N)</i>	Screen edits only To flood master-level flag (MASTRL) <i>NOTE: For the master-level flag only; can revisit if we need to also allow for sub-accts</i>	

Flood Type	Data Element to be Flooded	Required in the CU's File	Flood Field Options and (Format Validation)	Validation to Perform	Other Options on the Flood Screen?
B	SEG Code	ACCTBS	<i>New SEG Code (####)</i>	Validate against CU SEG/ Sponsor config (MNCNFD #2)	
B	Statement Mail Group Code	ACCTBS	<i>New Stmt Group code (0-9)</i>	Validate against Statement Mail Groups config (MNPRTC #12)	
B	Statement Style	ACCTBS	<i>New Stmt print style code (00-99)</i>	Validate against styles marked as "available" in the CU config (MNCNFE #21)	
B	User-defined Code 1, 2	ACCTBS	<i>New MLCOD1 (0-9)</i> <i>New MLCOD2 (0-9)</i>	Validate there's a description for this code in the CU config (MNCNFC #3)	
A	Wrong Email Address flag	ACCTBS	<i>Wrong email (Y/N)</i>	Screen edits only	Also write out a Tracker, same as PUPDINVEM

To Discuss

- What about updates to other misc. fields associated with a particular data element, like the "real" maintenance program does? (*i.e., enrolling in e-statements also needs to record the enrollment date and emp ID*)
- What about an "All memberships/accounts" option so they don't have to create an ACCTBS file if they want the flood done on every account they have?
- Anything to do in the non-member database? Closed members/accounts?
- Should we rank the above list based on how many times we have done this flood in the past, to prioritize which are most important?
- What about going through MNUPDT #1 and listing other fields a CU might want to flood? Worth the effort or wait for a request?

Ideas for Future Phases (Later)

Flood Type	Data Element to be Flooded	Required in the CU's File	Flood Field Options and (Format Validation)	Validation to Perform	Other Options on the Flood Screen?
A	eNotice Flag	ACCTBS	<i>New eNotice enrollment flag (Y/N) Send to flag (1=msg center only, 2=msg ctr+email) Send text alert (Y/N)</i>	Screen edits only What about requiring valid email address? Are there other things the enrollment feature does that this should also do? We've only done this one once (FILEHA) but had to reverse it becuz of some legal reason (ask Vilker?) – worth the trouble?	
B	eStatement Flag	ACCTBS	<i>New eStmt enrollment flag (0-3)</i>	Validate against what eStmt options they offer (MNCNFE #9) What about all the other things e-statement enrollment already does? What about requiring email address?	
??	ATM Limits / # of Trans (conditional on BIN, PIN/SIG, current amts - only flood if current limit is > \$xx)	No account list, just choose a BIN	<i>New \$ and # limits for all cards under that BIN</i>	There is a program on the EFT Toolbox that does this for all accounts under a certain BIN – since this wouldn't require a list of account numbers anyway, it would be better to turn that program into a member-facing one (if we want to do it at all)	
??	Code Word	??		Need to understand what this custom flood was used for	
??	Home Banking Flags	ACCTBS	<i>Allow access (Y/N) Other??</i>	Need to understand what this custom flood was used for	Maybe allow them to clear the Accepted use agreement date?
??	Loan Category (conditional on purpose/security codes)	??		Need to understand what this custom flood was used for	
??	Overdraft Protection Accounts	??		Need to understand what this custom flood was used for	
??	Print Additional Signers Flag	??		Not sure what this one would do	
??	Risk Score	ACCTBS ACTTYP New Risk Score	(New score would need to be in the CU file)	Actually this was simply a flood from OPENDLF to MEMBER5/6 – run after a soft-pull populated OPENDLF but didn't update MEMBER5/6 – should we just fix the soft-pull program to do that automatically instead??	

Flood Type	Data Element to be Flooded	Required in the CU's File	Flood Field Options and (Format Validation)	Validation to Perform	Other Options on the Flood Screen?
??	Tracker account suffix (conditional on Tracker Type)	??		Need to understand what this custom flood was used for	
??	Tracker Type / Memo Types	ACCTBS ACTTYP?? Old Tracker Type / Memo combination?	New Tracker Type / Memo combination This is one sitting in Track*IT right now; is it worth considering for a standard flood??	?? not sure I understand this one – did they create all of the new code combinations first and delete the old ones or something?	

New Master Flood Types Configuration

Based on the programs described in the following pages, would like to develop the behind-the-scenes structure and maintenance tools whereby we could expand the number of fields that can be flooded, without having to do additional development. Similar to how the 5300 feature lets us add new account codes and calculation routines without screen/program changes.

This should be designed by a programmer, but should allow for the definition of variables such as:

- Flood type code, description
- Data required in the account list (ACCTBS, ACCTBS+ACCTYP, ACCTBS+other text) and associated specs to be shown on the flood screen
- Table name(s) against which the CU's list of account numbers will be validated (MASTER, MEMBER1, MEMBER2, etc.)
- Table name(s) where the new data will be flooded (MASTER, MEMBER1, MEMBER2, etc.)
- Format for field being flooded (amount, date, Y/N, etc.)
- Possible values to be flooded, either a standard value like Y/N, 0/1, etc., or from values stored in another configuration table (or SYSCTL, which I'm sure will be a lot more complicated!)
- Possible options for the new value settings (such as a way to specify a value to flood, or leave the value alone, or pull it from the incoming file, etc. – see the samples on Page 19)
- Any required screen edits
- Any validation routines that should be run against the incoming data

I'm having trouble envisioning this, given the number of variations in our Phase 1 list of floods. I'm open to doing this a different way if we can build something that won't paint us into a corner when it comes to future expansion.

New Database Flood Engine

Screen 1 - Choose Flood Type

Access: TBD, see Page 23

Program: new

GOLD panel: new

Screen 1

*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8...+...9...+...0...+...1...+...2...+...3..

1	10/01/09	Flood Data to List of Member Accounts	14:40:14	Step 1: Choose Flood Type	UXXXXX-01
2	Search for description containing				
3	X>Select	Data Required in Account List			
4	Sel	Data Element to Be Updated	Acct Base	Acct Type	Other
5	-	1234567890123456789012345678901234567890	X	X	X
6	-	Additional Signers Delinquency Level	X	X	X
7	-	AFT Loan Payment Controls	X	X	
8	-	ANR Limits and Lock Flag (standard)	X		
9	-	CD Penalty Code	X	X	
10	-	Due Diligence Flag	X		
11	-	ECOA Code	X	X	
12	-	Electronic Hold Group	X		
13	-	Freeze Code	X	X	
14	-	Loan Assigned Employee ID(s)	X	X	
15	-	Loan Delinquency Fine Codes	X	X	
16	-	Loan Maturity and Review Dates	X	X	
17	-	Loan Payment Matrix	X	X	
18	-	Loan Purpose/Security Codes	X	X	
19	-	Loan Variable Rate Code	X	X	
20	-	Marketing Opt In/Out Flags	X		
21	-	Member Branch	X		
22	-	Photo ID on File Flag	X		
23	-	Reg E Opt Out	X		
24	-	SEG Code	X		
25				More...	
26	F7=CANCEL				
27					



Specifications

Field Name	Specifications, Comments, etc.
Data Element to be Updated	This is the list of whatever floods we have coded – see the table on Page 12
Data Required in Account List	Show an X in the column for the account base (all of them), account type (if applicable), and Other (if one or more additional fields will be used)

Screen 2 – Choose Flood Settings

*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8...+...9...+...0...+...1...+...2...+...3...

10/01/09 Flood Data to List of Member Accounts 14:40:14 Step 2: Choose Flood Settings UXXXX-01
 Flood type **Electronic Deposit Hold Group**
 Table name (in QUERYxx) for list of member accounts10
 New Value for All Accounts in the Account List
 Electronic deposit hold group .. ?

Your table must include these columns in this order
 1 ACCTBS (Account Base) 9 digits

F3=Backup F7=CANCEL Enter=Validate/Continue

Remember that all type “B” floods should offer a lookup to the configured values

Examples of how other types of floods should be handled on this screen:

Flood type **Additional Signers Delinquency Level**
 Table name (in QUERYxx) for list of member accounts10
 New Value for All Accounts in the Account List
 Additional signer delq level .. (0-4)

Your table must include these columns in this order
 1 Account base 9 digits
 2 Account type 3 digits
 3 Additional signer SSN 9 digits

NOTE: The (2) after the amount is just an indicator to GOLD of the decimal digit format so we don't have to use a ^ on the next line.

Flood type **ANR Limits and Lock Flag**
 Table name (in QUERYxx) for list of member accounts10
 New Value for All Accounts in the Account List
 Negative balance limit 1 1=No change 2=Specify (2) 3=Pull from my list
 Negative balance lock 1 1=No change 2=Y/Locked 3=N/Unlocked 4=Pull from my list

Your table must include these columns in this order
 1 Account base 9 digits
 2 Account type 3 digits
 3 Negative balance limit #,###,###,## (Opt)
 4 Negative balance lock Y or N (Opt)

Flood type **Loan Maturity and Review Dates**
 Table name (in QUERYxx) for list of member accounts10
 New Value for All Accounts in the Account List
 Loan maturity date 1 1=No change 2=Specify 3=Add .. months to date on file 4=Pull from my list
 Loan review date 1 1=No change 2=Specify 3=Pull from my list

Your table must include these columns in this order
 1 Account base 9 digits
 2 Account type 3 digits

Specifications

Field Name	Specifications, Comments, etc.
Flood type	Title of flood selected on screen 1
Table name (in QUERYxx) for list of member accounts	Entry required; upon Enter check for the presence of this file in QUERYxx

<i>Field Name</i>	<i>Specifications, Comments, etc.</i>
<p>New Value for All Accounts in the Account List</p>	<p>List the field name(s) that will be flooded, with either lookups to the appropriate configuration or a format notation such as (Y/N) or (CCYYMMDD) etc. Entry required</p> <p>In cases where there is more than one data element to be flooded, allow for a “No change” option, but just don’t allow to proceed if all fields are set to no change (at least one selection must be made)</p> <div data-bbox="667 369 1399 514" style="border: 1px solid red; padding: 5px; color: red;"> <p>It was tricky to define a layout that would fit all of the options even for just the sample ones above, so I’m open to ideas of better ways to present this, as long as they are clear to the user.</p> </div>
<p>Table must include these columns in this order</p>	<p>Would like to show which fields this particular flood expects; can use the official descriptions from the file being populated, or include that in the master config structure if that would be better (albeit redundant)</p> <p>Would also like to include the field length, as in “9 digits” or “3 characters” or “#,###,###.##” or whatever is appropriate</p> <div data-bbox="667 697 1399 808" style="border: 1px solid red; padding: 5px; color: red;"> <p>Programmer please advise if there is another way to handle this, but my idea is that we should provide on-screen documentation as to what the program expects</p> </div>

Screen 3 – Validate & Process

Programmer analysis will be needed on each individual flood type to see if we should also be updating other related fields, such as maintenance date/employee ID, etc., similar to the full-blown maintenance program. **Please advise** of any where there are additional steps that need to be done and we can discuss whether to include them in phase 1 or wait for phase 2.

```

*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8...+...9...+...0...+...1...+...2...+...3...
1 10/01/09   Flood Data to List of Member Accounts           14:40:14   Step 4: Validate Data and Process   UXXXX-01
2
3 # of records in your list   12,345,789
4 # of matches found         12,345,700 Compared to current open memberships (MASTER)
5 # that will be ignored     89
6
7 Run in A (A=Audit mode, P=Posting mode)
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26 F3=Backup  F5=Process  F7=CANCEL  F10=Re-validate List
27
  
```

- GOLD: should make this a pop-up window
- If P=Posting mode, **F5=Process** should present a warning message (call program CUTCCWDW) when using Posting mode:

```

1234567890123456789012345678901234567890123456
THIS WILL UPDATE LIVE MEMBERSHIP/ACCOUNT FILES
WARNING: THIS CANNOT BE UNDONE!
  
```

- If P=Posting mode, **write all changes to CUFMNT** along with the posting/exception reports
- F10=Re-validate List** should refresh the screen, re-reading the incoming list (that way if the # isn't right when they first access the screen, they can use a second session to refresh the file and then run the validation again, without having to start all over again)

Field Name	Specifications, Comments, etc.
# of records in your list	Show the total # of records in the file
# of matches found	Compare to either MASTER (if flood type uses ACCTBS only) or MEMBERx (if flood type uses ACCTYP, and according to flood type parameters, i.e., loans only vs. savings etc.) and show the # of hits Make the notation to the right of this number dynamic, so it might read: Compared to current open memberships (MASTER) OR Compared to current open accounts (MEMBERx)
# that will be ignored	The # of records that didn't get a hit

I don't know if we need to get any more granular than this, since there will be situations where we'll be checking against, say, both MEMBER5 and MEMBER6, or perhaps even all MEMBERx files. Please advise if this will be a problem.

Field Name	Specifications, Comments, etc.
Run in	A=Audit mode (generate reports but don't update data) P=Posting mode (update data and generate reports) should default to A and require an entry

Reports

Posting/Auditing

To show old and new values for every record updated by the flood.

- Sort by account base then type
- For the 'SET TO:' field you might also show "NO CHANGE" or "FROM ACCOUNT LIST" depending on what they selected on the screen
- Since some of these update multiple fields within the same flood routine, my thought was to break the group into sections for each value being updated. The sample below is for the ANR Limits and Lock Flag flood which has two values to potentially update.

```

*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8...+...9...+...0...+...1...+...2...+...3..
12/07/07 10:18.42                ABC TEST CREDIT UNION                LXXXXX                PAGE 1
RUN ON 12/07/07                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX                USER DAWNM
                                AUDIT ONLY

FLOOD TYPE: ANR LIMITS AND LOCK FLAG

VALUE UPDATED: NEGATIVE BALANCE LIMIT                SET TO: 03

ACCT BASE TYPE  OLD VALUE                NEW VALUE
3443 000 01                03
110 01                03
112 01                03
4997 110 01                03
4998 110 03                03
7888 110 00                03

VALUE UPDATED: NEGATIVE BALANCE LOCK                SET TO: Y/LOCKED

ACCT BASE TYPE  OLD VALUE                NEW VALUE
3443 000 Y                Y
110 Y                Y
112 N                Y
4997 110 N                Y
4998 110 N                Y
7888 110 Y                Y

# OF RECORDS IN FILE: 7
# OF MATCHES: 6 COMPARED TO: MEMBERX

```

Exceptions

To show any records in the CU's file that couldn't be updated (account # not found, suffix not found, or based on type "C" where the incoming data to be flooded didn't adhere to validation standards used by the flood program).

```

*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8...+...9...+...0...+...1...+...2...+...3..
12/07/07 10:18.42                ABC TEST CREDIT UNION                LXXXXX                PAGE 1
RUN ON 12/07/07                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX                USER DAWNM
                                AUDIT ONLY

FLOOD TYPE: ANR LIMITS AND LOCK FLAG
# OF RECORDS IN FILE: 7
RECORDS NOT MATCHED: 1 COMPARED TO: MEMBERX

ACCT BASE TYPE  REASON
234 112  ACCOUNT NOT FOUND

```

- The only other reasons I can think of here besides not finding a match to an account in MASTER is for flood type "C" where the data in the incoming file did not pass the validation tests to allow it to be entered into the database. You could create messages specific to the error (like "AMOUNT TOO LARGE" or "MATCHING VALUE NOT FOUND IN CONFIGURATION") but I'm okay if you want to just use a generic message no matter what the validation problem was, such as: **INVALID DATA IN FILE**

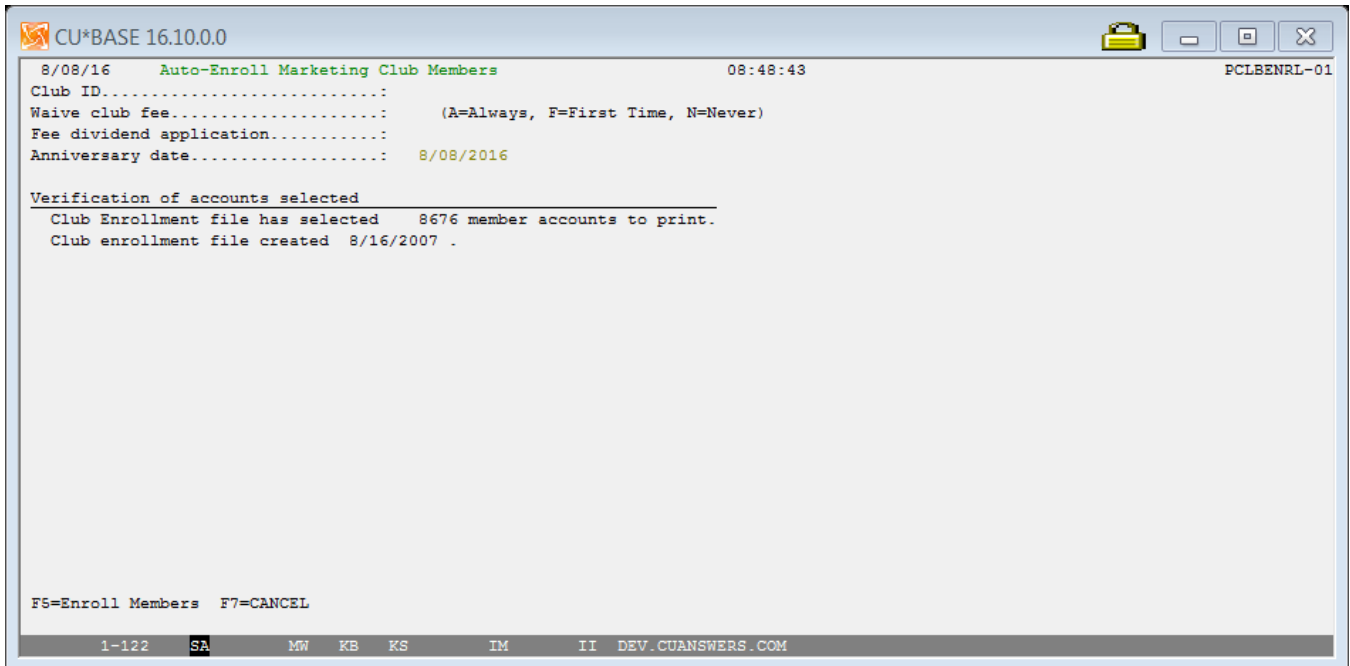
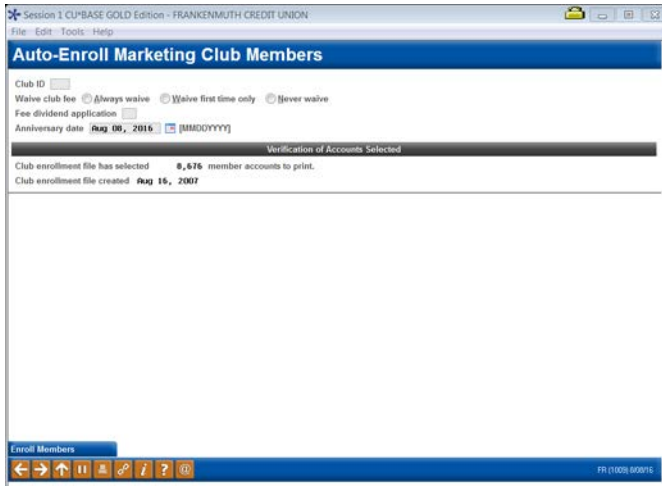
Examples of Existing Flood Programs

This section is primarily for info only, and maybe to serve as inspiration for the programmer as to user-defined flood techniques we've already developed. Someday we might enhance some of these tools, but **at the very least I want to include in the doc a list of all of the ways a CU can manipulate data for batches of accounts.** Some of these are very powerful tools and we should get some credit for them as a body of work.

MNMRKT #7 Auto-Enroll Marketing Club Mbrs

PCLBENRLCL

GOLD Panel: 1009

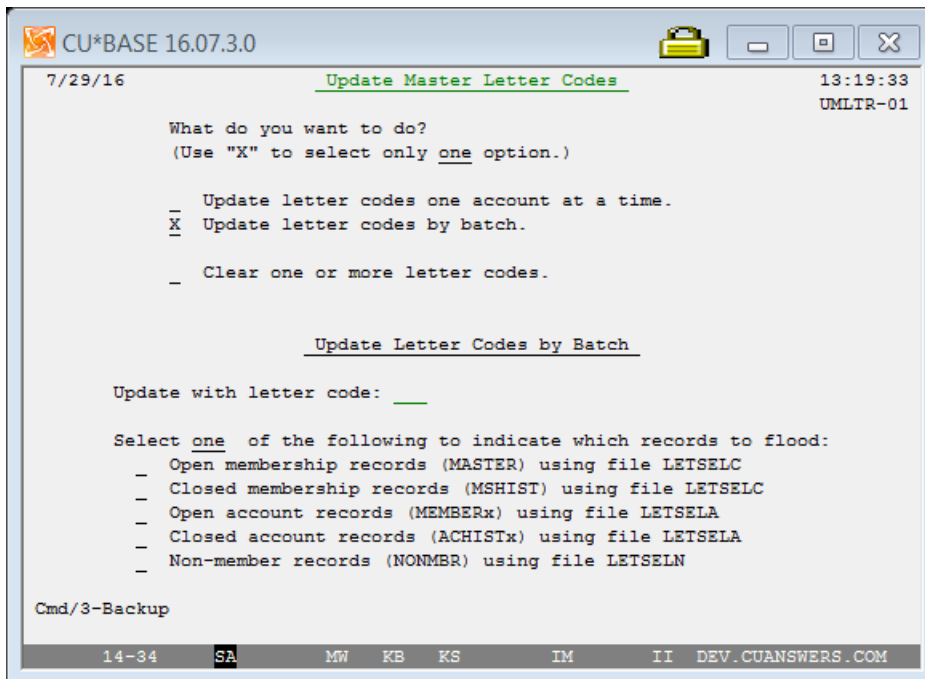
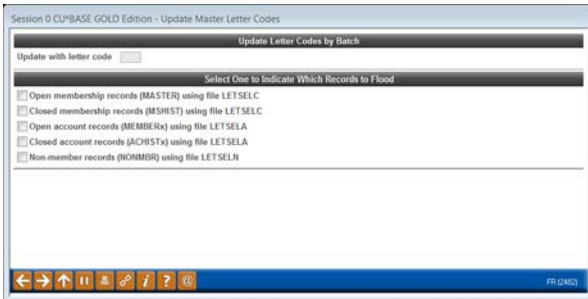


MNMRKT # 10 Tag Mbr Accts with Letter Codes

UMLTRCL

GOLD panel: 2482

(Select "Update Letter Codes by Batch" from the initial panel, also 2482)



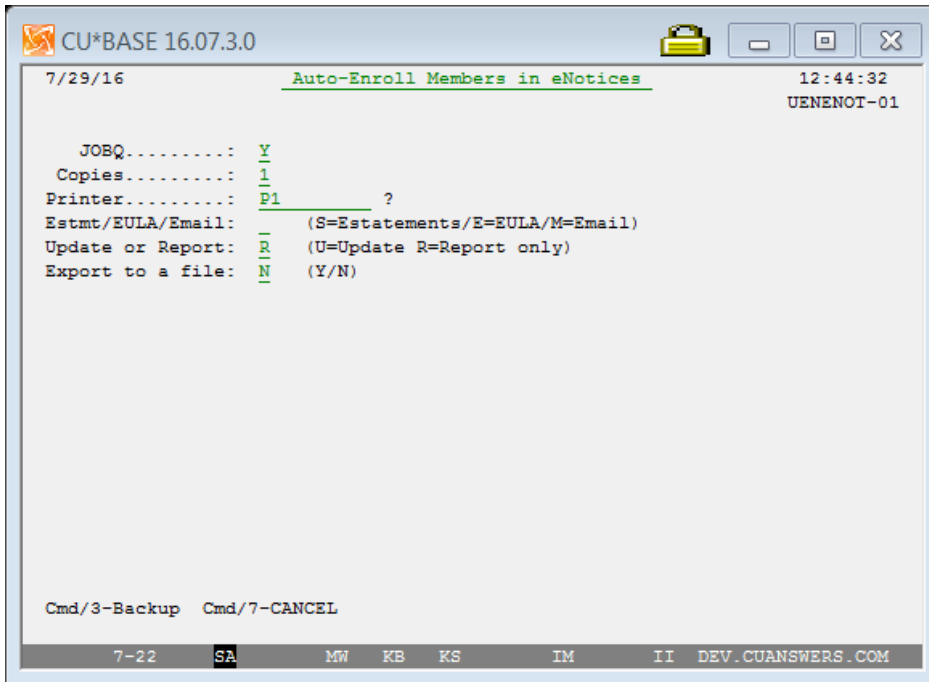
MNOP21 # 1 Flood e-Stmt Mbrs for e-Notices

UENENOTSCL

GOLD panel: 3448



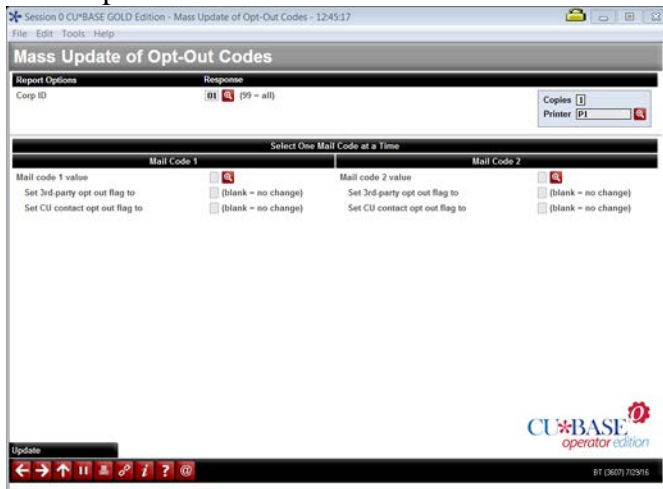
Could we make this one client-facing by improving the field labels on this screen??



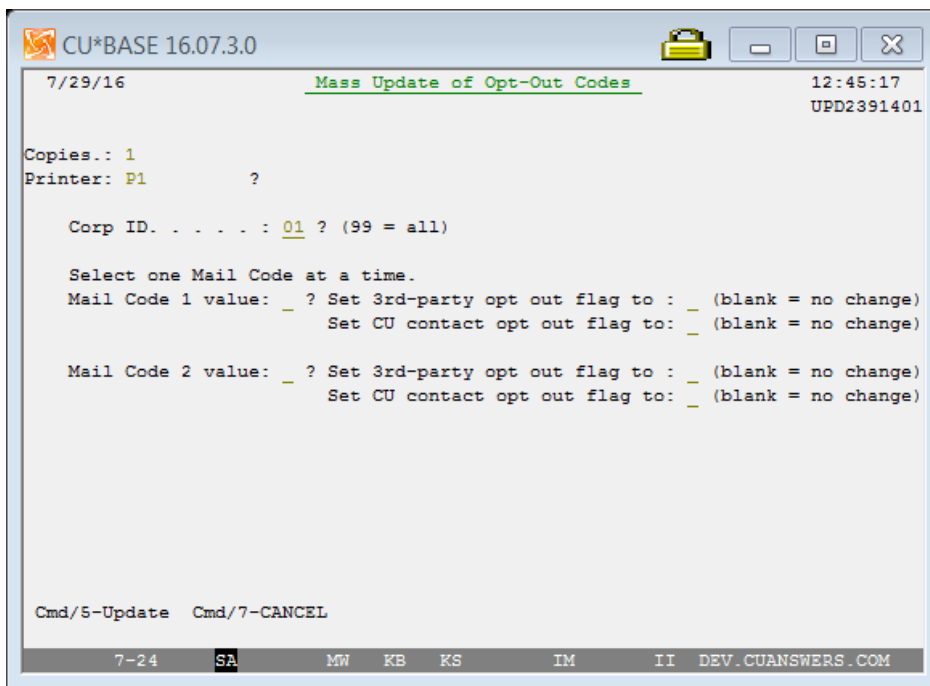
MNOP21 # 2 Flood Mail Code to Opt Out Flag

UPD23914CL

GOLD panel: 3607



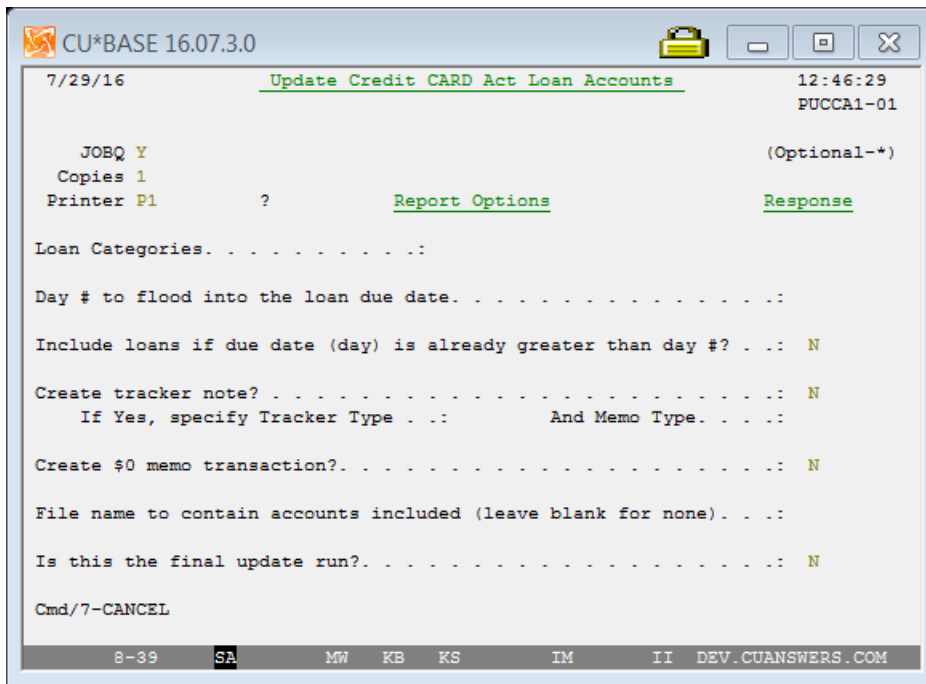
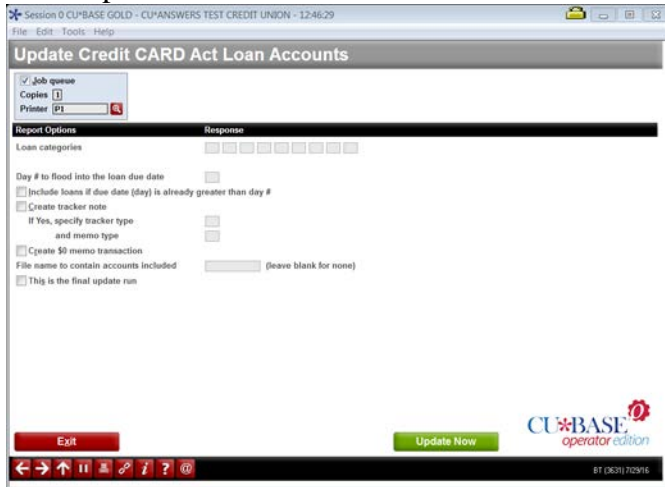
Is this one even worth keeping any longer?



MNOP21 # 3 Credit CARD Act--Flood Due Dates

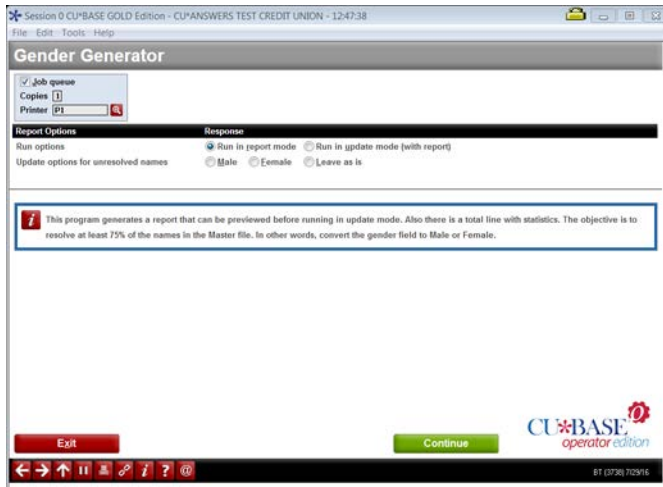
PUCCASCCL

GOLD panel: 3631

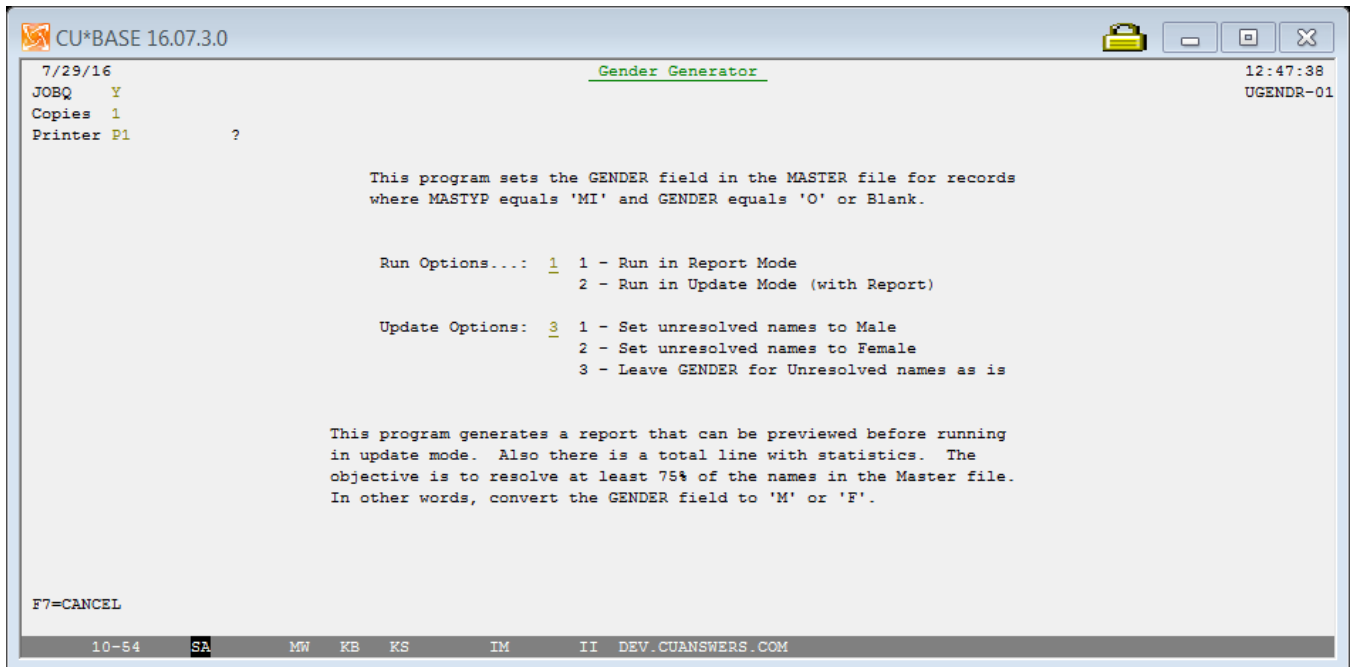


MNOP21 # 4 Gender Generator

UGENDRSCCL
GOLD panel: 3738



Could we make this one client-facing by improving the field labels on this screen??



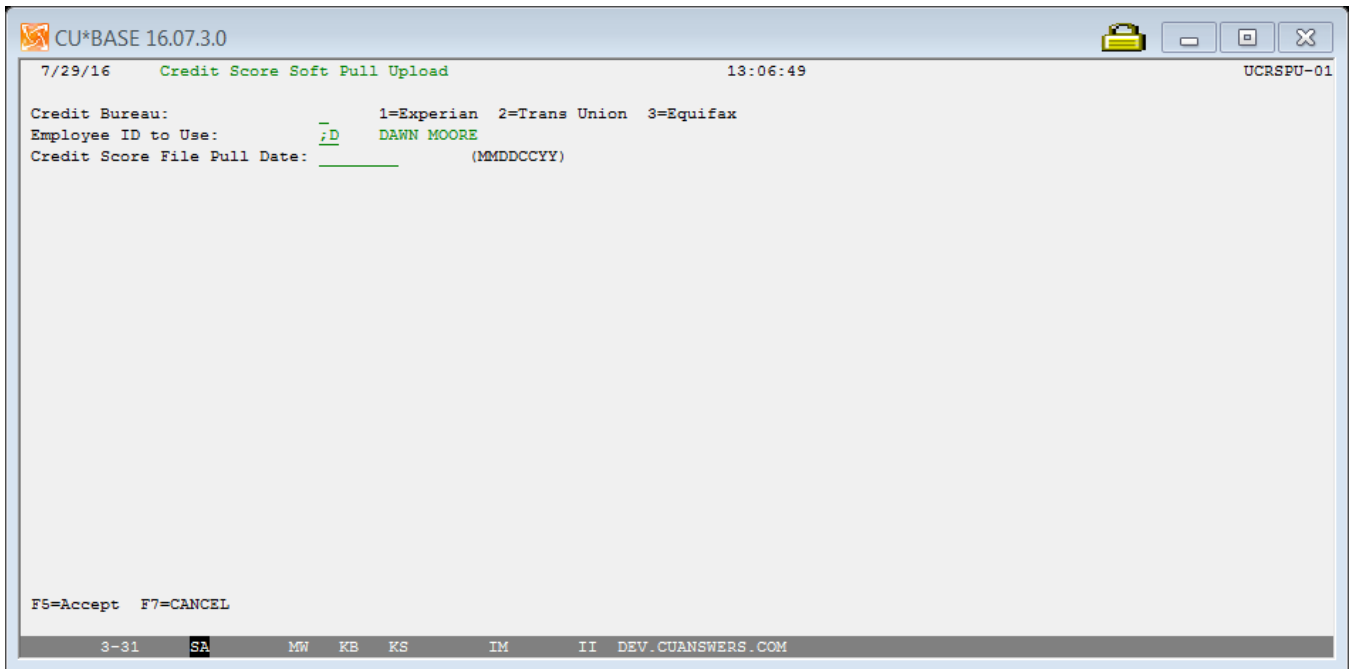
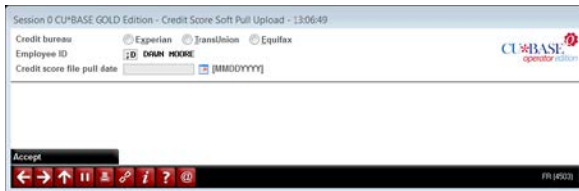
MNOP21 # 5 Open End Contract Flood

CNV27600CA
(unable to launch this on DEV)

MNOP21 # 9 Credit Score Soft Pull - Upload

UCRSPUCL

GOLD panel: 4503



Reassign Tracker Follow-ups

(Coming in Proj# 42412)

This project adds new Tracker feature for reassigning follow-ups. This would allow the credit union to transfer follow-ups to another employee when an employee leaves the credit union. This is a recurring flood request that programming receives (see PS#42140 as an example).

Wrong Email Address Flag & Marketing Opt In/Out Flags

(Project# TBD)

We are currently writing a spec for a similar type of flood for Xtend Member Reach, so they can bring in a file of just email addresses, and have the system match that against an account base and then change one or the other of these flags:

- Wrong email address Y/N

OR

- 3rd Party Marketing Opt In/Out
- CU Marketing Opt In/Out

That tool will also include a feature that produces a database file in QUERYxx that just contains the account bases that were matches against the email addresses in the original list. The CU could then use that list in the generic flood tool for other purposes.

Others currently in Toolbox??

Update PANCARD limits? UPANLMSCCL

Others?

Menu Changes

Menu changes are done by Dawn (and Bob C.) with the release. **Programmer, please add new programs to MNNEW for testing** and so that Dawn has a call to use when menu changes are done.

- Add new master config options to MNOP12 or ??
- Add new flood tool to MNFILE